

# Project Report: Comfy Online Store

---

## Overview

With smooth PayPal connection, the Comfy Online Store project is a feature-rich e-commerce web application that makes product purchases easier. The project's essential elements are described in this report, including the database schema, front-end and back-end architecture, Docker setup, and API endpoints. From product browsing to transaction completion, the initiative is meant to give users a seamless and safe purchasing experience.

---

## Front-End Components

The Comfy Online Store's front end is constructed with HTML, CSS, and JavaScript. Important elements consist of:

### Webpages

#### 1. Homepage (index.html):

- Contains connections to other pages, a call-to-action button in the hero area, and a navigation bar.
- A sidebar is included for convenient navigation, and Font Awesome is used for the icons.

#### 2. The "about" page (about.html):

- Offers background and mission information about the shop.
- Styled for a unified user experience with a nod to the homepage.

#### 3. Products Page (products.html):

- Provides a list of all items that are offered along with the ability to search for products by name.
- Shows dynamically retrieved product names, prices, and photos from the back end.

#### 4. Single Product Page (product.html):

- Provides comprehensive details about a single product, such as a picture, a description, and a cost.
- PayPal is integrated for payment processing; buttons are provided to start transactions and to see modals that indicate errors or successes.

#### 5. Dashboard (dashboard.html):

- The dashboard permits addition of new items and provides order history information.
- Add product forms contain the following fields: name, price, description, image, company.

## Styles and Scripts

### CSS (styles.css):

- **For all pages** it defines their layout and design including use of the responsive design on different screen sizes.
- It has styles for navigation bars, buttons, forms, modals and error messages.

### JavaScript:

- index.js, products.js, product.js, dashboard.js: These files help manage dynamic content loading and form submission as well as assist in interacting with the PayPal API.
  - toggleSidebar.js: This file executes opening and closing actions towards sidebar.
- 

## Back-End Components

The back-end of Comfy Online Store is established on Node.js and Express that gives RESTful API endpoints for managing products and orders. Some of the major components are:

### Controllers

#### Product Controller (productController.js):

- It handles product-related activities such as creating, getting, or uploading images.
- The controllers interact with MongoDB to save or fetch data on a product.

#### Order Controller (orderController.js):

- It has methods to manage orders such as create and capture.
- This will connect to PayPal SDK enabling payments processing.
- Data input validation, order states control like failed, complete or any other forms part of this class.

## Routes

### 1. Product Routes (`productRoutes.js`):

Endpoints for product operations:

- POST REQUEST: `/api/v1/products`
- GET REQUEST: `/api/v1/products`
- GET REQUEST: `/api/v1/products/:id`
- POST REQUEST: `/api/v1/products/uploadImage`

### 2. Order Routes (`orderRoutes.js`):

- Defines endpoints for order operations:
  - POST REQUEST: `/api/v1/orders`
  - POST REQUEST: `/api/v1/orders/capture`
  - GET REQUEST: `/api/v1/orders`
  - GET REQUEST: `/api/v1/orders/config/paypal`

## Middleware

- **Error Handling:**

- `notFoundMiddleware.js`: Handles 404 errors for any routes that are not defined.
- `errorHandlerMiddleware.js`: Handles global errors and send appropriate responses.

- **Security:**

- `helmet`: Puts security headers.
- `xss-clean`: Clean user input to prevent XSS attacks.
- `cors`: Enables cross-site scripting sharing.
- `express-mongo-sanitize`: Sanitize MongoDB queries to avoid NoSQL infection.

## Database

- **MongoDB:**
    - **Product Schema (Product.js):** Gives the structure of product documents which consist of fields such as name, price, image and company.
    - **Order Schema (Order.js):** Describes how an order document should look like i.e. it must have fields known as product name, price, amount and payment status.
- 

## Docker Setup

To containerize the application, the following Docker file and setup instructions are used:

### Docker file:

```
FROM node:18
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 5000
CMD ["npm", "start"]
```

### Instructions For the Setup of Docker:

1. **Build the Image File of Comfy Store:**

```
docker build -t comfy-store .
```

2. **Run the Container of Comfy Store:**

```
docker run -p 5000:5000 comfy-store
```

3. **Open the Comfy Store at:**

Now open a browser and navigate to <http://localhost:5000>.

---

## Database Schema

The database schema for the Comfy Online Store uses Mongoose, which has two main models: the Product Model and the Order Model.

## Product Model (Product.js)

- **Fields:**
  - name: A required string with a maximum length of 100;
  - price: A number that is required;
  - description: A string that is required with a maximum length of 1000;
  - image: A default string value equals to a placeholder image;
  - company: String. It must be one of its predetermined values such as 'ikea', 'liddy' and 'marcos'.

## Order Model (Order.js)

- **Fields:**
  - name: The name field is always required and should be input by the user.
  - price: The price field is always required and should be input by the user.
  - amount: The amount field is always required and should be input by the user.

## Source Code

The source code for Comfy Online Store can be found in GitHub. To access it follow this link

[Comfy Online Store GitHub Repository](#)

## PayPal Credentials for Testing Account

- Email: sb-ca0lm31552190@personal.example.com
- Password: >fX\*8/vc